Noname manuscript No.
(will be inserted by the editor)

A novel unified deep neural networks methodology for *use by* date recognition in retail food package image

Liyun Gong · Mamatha Thota · Miao Yu · Wenting Duan · Mark Swainson · Xujiong Ye · Stefanos Kollias

Received: date / Accepted: date

Abstract There exist various types of information on retail food packages, including *use by* date, food product name and so on. The correct coding of *use by* dates on food packages is vitally important for avoiding potential health risks to customers caused by erroneous mislabelling of *use by* dates. It is extremely tedious and laborious to check the *use by* dates coding manually by a human operator, which is prone to generate errors thus an automatic system for validating the correctness of the coding of *use by* dates is needed. In order to construct such a system, firstly it needs to correctly automatic recognise *use by* dates on food packages. In this work, we propose a novel dual deep neural networks based methodology for automatic recognition of *use by* dates in food package photos recorded by a camera, which is a combination of two networks: a fully convolutional network (FCN) for *use by* date ROI detection and a convolutional recurrent neuron network (CRNN) for date character recognition. The proposed methodology is the first attempt to apply deep learning for automatic *use by* date recognition. From comprehensive experimental evaluations, it is shown that the proposed method can achieve high accuracies in *use by* date recognition (more than 95% on our testing dataset), given food package images with varying lighting conditions, poor printing quality and varied textual/pictorial contents collected from multiple real retailer sites.

Liyun Gong, Mamatha Thota, Miao Yu, Wenting Duan, Xujiong Ye and Stefanos Kollias
School of Computer Science, University of Lincoln, UK
E-mail: yun200634234@outlook.com
E-mail: myu,wduan,xye,skollias@lincoln.ac.uk

Mark Swainson
National Centre for Food Manufacturing, Holbeach Technology Park, UK E-mail: mswainson@lincoln.ac.uk

Keywords expiry date recognition · food security · deep learning · fully convolutional network (FCN) · convolutional recurrent neuron network (CRNN)

1 Introduction

Food production is the largest manufacturing sector in the European Union. It is with a reported turnover of 945 billion [1] and accounts for 13.3% of the total EU-28 manufacturing sector. While the availability of food still remains a primary concern in developing countries, food quality/safety is another very important issue in more developed societies. The requirement of food safety is essential across all food supply chains, which is underpinned by food science and technology and assured by a combination of procedures and operational control systems, such as Hazard Analysis Critical Control Point (HACCP) [2].

The food product information printed on the food package is a important for the food safety. Incorrectly labelled product information on food packages such as the expiry date can cause food safety incidents like food poisoning due to the consuming of food product with its genuine *use by* date being expired. Moreover, such faults/issues will incur high reputation and financial cost to food manufacturers while may result in product recalls. Due to the aforementioned negative effects caused by incorrectly labelling/coding of *use by* date on food packages, the verification of the correctness of the *use by* date printed on food packages is important. With respect to the *use by* date information verification, the following two steps are followed: i). *use by* date recognition: the printed *use by* date on a food package is identified ii). correctness verification: identified *use by* date information is compared with the ground-truth

one that is stored in the database for the verification purpose

The *use by* date recognition is an indispensable step for verifying the correctness of the *use by* date information printed on food packages. Traditionally, this step is done by a human operator who manually picks a food package for the inspection of *use by* date, which creates mundane and repetitive tasks thus placing the human operator in an error-prone working environment. To avoid errors caused by the human operator and save human labors, optical character recognition (OCR) systems [3] can be applied to automatically recognize *use by* date characters based on captured food package images taken by RGB cameras. However, the existing OCR systems work only effectively for recognizing clear characters in high quality images with uncomplicated backgrounds while requiring the characters consistency with respect to format and viewing angle. This limits the applications of OCR systems for recognizing *use by* date in the real world scenarios, where there exist high variabilities of *use by* date characters fonts/angles, complicated food packaging designs with rich colours/textures information, blurred characters on images, captured food package images with low qualities due to poor lighting conditions in a food manufacture/retailer site and other challenging factors.

Considering the limitations of traditional OCR systems, we target to detect/recognise *use by* date 'from the wild' from food package images, that is, to be able to detect/recognise *use by* date from a variety of packages with different characters formats under challenging conditions (including but not limited to low image qualities, blurred characters, complex image background, etc.) in real food manufacturing/retailor sites. Currently there are no such research works for detecting/recognising *use by* date 'from the wild', but there do exit related works for detecting/recogsing texts from natural scenes.

With respect to texts detection in images captured from different types of natural scenes, traditional image processing based methods such as Stroke Width Transform (SWT) [4] and Maximally Stable Extremal Regions (MSER) [5] have been applied. In recent years, deep learning based approaches have been widely applied for texts detection 'from the wild'. In [6] and [7], different deep neural network models have been developed to automatically learn effective features for texts detection under a variety of scenes. However, these methods consist of several stages to detect texts, which are probably sub-optimal and time consuming. To overcome related limitations, a light-weight fully convolutional network (FCN) based approach is proposed in [8], which achieves a higher detection accuracy with minimum number of processing stages (i.e., low computational costs).

From detected regions of interests (ROIs) containing only texts, characters/words in the ROIs can then be recognized by certain text recognition techniques. Traditionally some description features such as SIFT, HOG or DPM are extracted, based on which certain classifiers such as support vector machine (SVM) or artificial neural network [9] are applied for recognising texts. Deep learning based approaches have also been developed for texts recognition. In [10], a convolutional neural network (CNN) is applied for extracting most representative features applied for recognizing characters/words in texts. In [11], a convolutional recurrent neural network (CRNN) composing of both convolutional layers and recurrent layers, is applied for texts recognition. Representative features are extracted by convolutional kernels in the convolutional layers as in [10]. Moreover, the contextual information in texts is considered and modelled by recurrent layers in the CRNN for a more accurate recognition purpose. More complicated deep learning based four-stage text recognition methods are proposed in [12], which exploit spatial transformer network for normalizing text images and more complicated ResNet backbone for features extraction. Besides, both the connectionist temporal classification (CTC) or attention-based sequence prediction (Attn) schemes are evaluated for estimating the output character sequence from identified image features.

In this work, we adapt the traditional deep neural networks originally developed for detecting/regonizing the texts, to a new domain for detecting/recognising of *use by* date on food package. In specific, the FCN proposed in [8] and CRNN in [11], which are both lightweighted and achieve good performance for texts detection/recognition 'from the wild', are fine-tuned and combined together to detect/recognise *use by* date. Loss functions related to our tasks of *use by* date detection/recognition are defined and minimized to fine-tune FCN and CRNN network parameters, to adapt the original 'text' detection/recognition networks to a new '*use by* date' detection/recognition ones, based on a training dataset consisting of food package images. This procedure of networks fine-tuning is also relevant to transfer learning [13], which adapts a trained model (e.g., a deep neural network) obtained for a source domain task for performing different but relevant target domain tasks. With respect to this work, the source domain task is text detection/recognition and the target domain task is *use by* date detection/recognition. The obtained fine-tuned network can effectively detect/recognize *use by* date information on food packages images with different colours/textures under different image qualities. From

comprehensive evaluations, it is shown that the proposed method achieves high accuracies and outperforms other deep learning based methods for *use by* date detection/recognition. The structure of this paper is as follows: The details of the proposed deep neural network based *use by* date detection/regognition are proposed in Section II. Related experimental evaluations for *use by* date detection/recognition on food package images are presented in Section III. Conclusions and future works are given out in Section IV.

2 The Proposed Approach

In this work a deep learning based system is devised, for a robust solution for *use by* date recognition on food packages under various of challenges (e.g., varying lighting conditions, poor printing quality, diversities of colours/textures/texts on food packages, etc.). The structure of the proposed system is in Fig. 1, which is a fusion of two networks. The first one is a fine tuned fully convolutional network (FCN) as in [8], which is responsible for the detection of the ROI of *use by* date. It acts as a filter to identify the image patch including *use by* date information from a whole food package photo, making the recognition task be performed on that specific small image patch instead of the whole image region to reduce the computational cost. Besides, it can avoid recognition errors caused by interferences from other texts on a variety of parts on a food package image by restricting only on the detected ROIs. The second network is a fine tunned convolutional recurrent neural network (CRNN) [11], which is used for the date characters recognition based on the image patch obtained from the first network. Multiple levels of features are extracted from equally divided regions in the *use by* date ROI to recognize date characters within each region, while the contextual relationships between characters in consecutive regions (e.g., the character V will follow NO for composing NOV) are also considered and modelled by the recurrent layers of CRNN. Based on a relatively small training dataset of food package images, these two networks are fine-tuned for detecting and recognizing date information on the food package.

2.1 Fully Convolutional Neural Network for *use by* date detection

The fully convolutional network (FCN) in [8] was developed for texts detection. Its architecture is shown in the left dash rectangle box Fig. 1, which composes of three parts: feature extractor stem part, feature-merging branch part and output part.

The main architecture of the feature extractor stem part is a PVANet [14], with composing of interleaving convolution and pooling layers. Four levels of feature maps, denoted as f_i are extracted from the original input image from convolutional layer, whose sizes are $\frac{1}{32}$, $\frac{1}{16}$, $\frac{1}{8}$, $\frac{1}{4}$ of the original input image. Multi-scale feature maps can enable detection of text regions with different sizes. Convolutional kernels in convolutional layers are applied for estimating feature maps. In specific, the kth feature map in the lth convolutional layer of the feature extractor stem part, denoted as a_k^l can be calculated by the kth convolutional kernel associated with the lth convolution layer, denoted as W_k^l as:

$$\begin{aligned} net_k^l &= W_k^l \otimes a^{l-1} \\ a_k^l &= f(net_k^l) \end{aligned} \tag{1}$$

where a^{l-1} represents the feature map in the $l-1$th layer, net_k^l represents the net input related to the kth feature map in the lth layer, $f(\cdot)$ is an activation function (e.g., sigmoid, ReLu, Tanh, etc.) and $\otimes$ represents the cross production operator. Moreover, the PVANet in the feature extractor stem part also includes pooling layers, which down-sample the feature maps by a factor of 2.

From the feature extractor stem part, four feature maps ($f_1 - f_4$) are extracted which are then merged to obtain a feature map in the feature-merging branch part through the following way as in [8]:

$$\begin{aligned} g_i &= \begin{cases} unpool(h_i) & if \quad i \leq 3 \\ conv_{3\times3}(h_i) & if \quad i = 4 \end{cases} \\ h_i &= \begin{cases} f_i & if \quad i = 1 \\ conv_{3\times3}(conv_{1\times1}([g_{i-1}; f_i])) & if \quad i = 4 \end{cases} \end{aligned} \tag{2}$$

where h_i is the feature map after the i-th merging stage (i=1,2,3,4). The operator [;], represents concatenation of tensor elements. At the i-th merging stage, the feature map obtained at the last stage h_{i-1} is firstly fed to an unpooling layer for doubling the size (g_i), then it is concatenated with the feature map f_i to merge into a new feature map. A $conv_{1\times1}$ operator is applied to cut down the channel numbers to reduce computation, followed by a $conv_{3\times3}$ to finally produce the merging stage output. At the end of the feature map feature-merging branch part, a $conv_{3\times3}$ layer based on h_4 produces the final merging output g_4 and feeds it to the output layer.

Multiple $conv_{1\times1}$ operations are applied in the final output layer, which converts the merging output with 32 channels into the following three output maps:

i). a score map F_s with one element in it representing the likelihood that the element belongs to the *use by* date region.

ii). a four-channel geometry map F_g, with each of its element containing the estimated horizontal/vertical

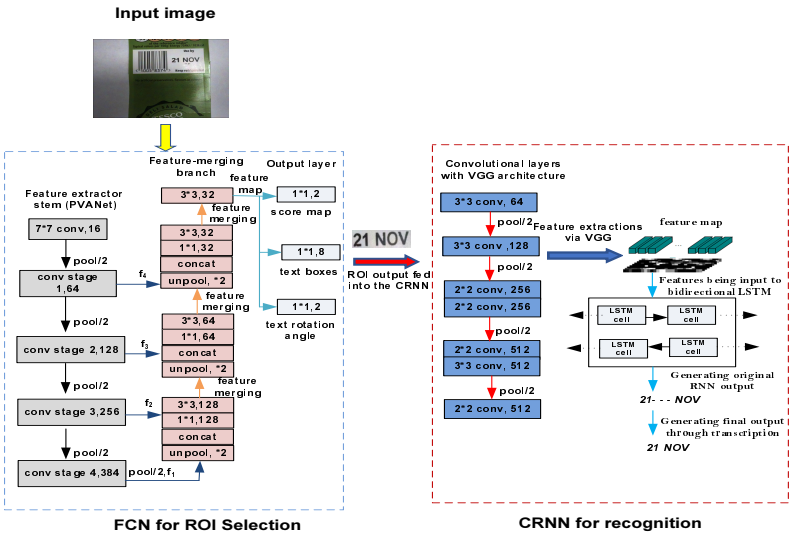

Fig. 1 Architecture sketch of the *use by* date recognition system

distances from the current element position to the top-right and bottom-left corners of a candidate rectangle *use by* date region

iii). a angle map F_a, with each of its element representing the estimated angle of a candidate rectangle *use by* date region

Based on obtained outputs F_s, F_g and F_a, multiple candidate *use by* date regions can be obtained. These regions are finally merged by the non-maximum suppression (NMS) methodology as in [8] to obtain the final output. The related text detection results are illustrated in Fig. 2.

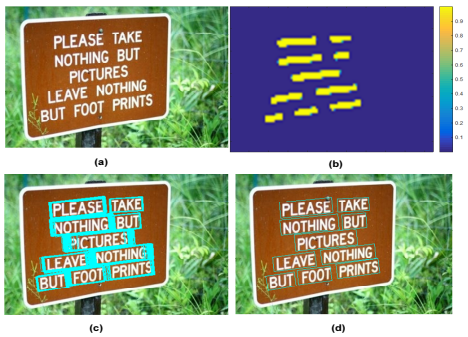

Fig. 2 The exemplified text region detection results. (a) Original image (b) Score map output by the FCN (c) Candidate rectangle boxes of text regions before NMS (d) Text detection results after NMS.

2.1.1 Fine tuning of Fully Convolutional Neural Network

In order to fine-tune the FCN to adapt the original text detection network, to a *use by* date detection network, we prepare a training dataset containing a variety of food package images. For each image in the training dataset, we acquire four points enclosing the *use by* date region and save their positions in a txt file (as in Fig. 3). Based on all training images and associated saved positions, a loss function for fine-tuning the FCN is defined as:

$$L_{FCN} = \sum_{i=1}^{N} L_s^i + \lambda L_g^i \tag{3}$$

where N is the number of images in the training dataset. L_s^i and L_g^i represent score/geometry losses as in respectively [8] corresponding to the $i-th$ image, while λ is a balancing parameter of the two. The term L_s^i is defined as:

$$L_s^i = -\beta Y_i^* log\hat{Y}_i - (1-\beta)(1-Y_i^*)log(1-\hat{Y}_i) \tag{4}$$

where $\hat{Y}_i$ represent the predicted score map for the ith image. And Y_i^* represents the groundtruth one, with value 1s being in the *use by* date region enclosed by boundaries determined by four points as in Fig. 3 and 0s elsewhere. β is a balancing parameter. While the L_g is defined as scale-invariant Intersection over Union (IoU) loss as in [8]. In this work, ADAM algorithm [17] is applied to fine-tune the weights on the FCN for minimizing the loss function in (3).

After fine-tuning, the original text detection network is converted into a *use by* date detection one, as illustrated in Fig. 4.

Fig. 3 The four points enclosing the *use by* date region and their positions saved in a txt file

Fig. 4 FCN detection results before/after the fine tuning. (a). Text region detection results before the fine-tuning (b). Date code regions detection results by the FCN after the fine tuning

2.2 Convolutional Recurrent Neural Network for *use by* date recognition

The second part of the proposed *use by* date recognition system is a CRNN which was originally developed for image-based text sequence recognition, as shown in the right dash rectangle box in Fig. 2. The CRNN is a light-weighted network for text recognition, which is mainly composed of three parts including the feature extraction part, the bidirectional LSTM-RNN part and a transcription layer part.

The feature extraction part follows a VGG architecture as in [15]. An input image is divided into T different image patches, while feature vectors $x = \mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_T$ corresponding to T different patches in the image (as illustrated in Fig. 5) are extracted through convolutional layers and pooling layers in the feature extraction part. Extracted feature vectors from the feature extraction part is then fed into a deep bidirectional Recurrent Neural Network (RNN) with LSTM unit (as the bidirectional LSTM in Fig. 2), to predict the label distribution denoted as $y = \mathbf{y}_1, \mathbf{y}_2, ..., \mathbf{y}_T$. Recurrent layers in this bidirectional LSTM-RNN model captures the contextual dependencies between texts information in consecutive image patches for a more stable and accurate text characters recognition. Another advantage of the bidirectional LSTM-RNN is that it is able to operate on arbitrary lengths text sequences, which makes it be suitable for this work to recognize different lengths of *use by* date in different formats (both DD/MM and DD/MM/YY).

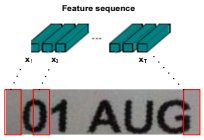

Fig. 5 Different image patches (enclosed by red rectangles and corresponding extracted features).

The bidirectional LSTM-RNN is composed of LSTM units, which are used to solve the vanishing gradient problem existing in the traditional RNN which limits the range of context that RNN can process. A LSTM unit consists of a memory cell and three multiplicative gates, which are the input, output and forget gates. The detailed structure of the LSTM unit is shown in Fig. 6. Based on the current input $\mathbf{x}_t$ for the t-th LSTM unit and previous LSTM memory state $\mathbf{c}_{t-1}$, the final LSTM output $\mathbf{h}_t$ can be derived according to the related arithmetic operations as detailed in [16].

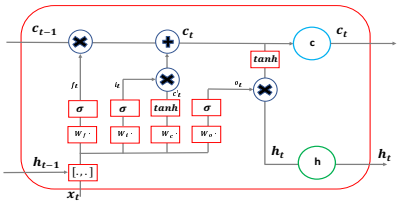

Fig. 6 The structure of a LSTM unit.

The final layer in the CRNN is for transcription, which converts the predictions made by the second bidirectional LSTM-RNN into a label sequence l, by maximizing a conditional probability given the bidirectional LSTM-RNN predictions $(y = \mathbf{y}_1, \mathbf{y}_2, ..., \mathbf{y}_T)$ as defined in (5).

$$p(l|y) = \sum_{\pi:\beta(\pi)=l} p(\pi|y) \tag{5}$$

where π represents a characters sequence, the function $\beta(\cdot)$ maps π to l by removing spaces and repetitive characters as in [11]. The probability $p(\pi|y)$ is defined as $p(\pi|y) = \Pi_{t=1}^{T} \mathbf{y}_t(\pi_t)$, where T is the number of bidirectional LSTM-RNN outputs and $\mathbf{y}_t(\pi_t)$ represents the probability of a character π_t in π given the predicted output distribution $\mathbf{y}_t$ from the tth image patch.

2.2.1 Fine-tuning of CRNN

For fine-tuning the CRNN to make it recognize the *use by* date, a loss function L_{CRNN} is defined as in (6):

$$L_{CRNN} = \sum_i p(l_i|y_i) \tag{6}$$

where l_i and y_i represent the groundtruth label for *use by* date the and the CRNN prediction for the ith image respectively. Based on the above loss function, the stochastic gradient descent (SGD) algorithm is applied in this work to fine tune the CRNN weights.

3 Experimental study

In the experimental section, we provide evaluation results of the proposed system for *use by* date recognitions, based on food package images collected from different retailer sites.

3.1 Collected food package images

We have collaborated with OAL, a leading food company in Lincoln, UK, for collecting a variety of food package images from different superstore sites. Extremely blurred images which are excluded and totally a number of 2424 images are chosen for evaluation. Representative images are shown in Fig. 7

Fig. 7 Representative food package images at different superstore sites.

Based on the collected images, we divide them into two parts. 70% of images are used as training dataset for fine-tuning networks and 30% of images are used as a testing dataset for performance evaluating. Experiments are carried out on a GPU-supported Server, with two NVIDIA Tesla P100 GPUs.

3.2 FCN evaluation

The detailed structure of the FCN is shown in detailed in the left dash rectangle in Fig. 1. We have fine tuned the FCN by the Adam algorithm implemented in python with tensorflow library. To speed up learning, we uniformly sample 512x512 crops from images to form a minibatch of size 24. The fine-tuned network is tested on captured images of different food packages from different retailer sites. Related results are presented in Fig. 9 and 10, which show that the proposed network can successfully detect expiry date on different types of images, even under challenging situations such as poor lighting conditions and low printing qualities.

Fig. 8 *use by* date detection results on clear images with different colours/textures

Fig. 9 *use by* date detection results on images with poor qualities (missing parts, blurred, etc.)

Quantitatively, we evaluate the accuracy of the text detection, results are presented as in Table 1, which shows the performance of the FCN under different iteration values of the Adam algorithm. We can see that the FCN achieves the best performance after 100000 iterations.

Moreover, we've compared the performance of the fine tuned FCN with other fine tuned popular deep neu-

Table 1 *use by* date detection performance via FCN.

Iteration	Miss detection	False alarm	Accuracy
5000	3.34%	0.696%	96.4%
9000	3.34%	0.139%	96.5%
50000	1.39%	0.975%	97.6%
100000	1.67%	0.279%	98.2%
150000	1.67%	0.557%	97.6%

ral networks such as CTPN [6]and Seglink [7]for *use by* date detection with related results being provided in Table 2. For fair comparisons, the same training algorihtm (Adam) [17] is applied and related training parameters (such as iteration number, learning rate, etc.) are chosen to be the same. From the results, it is shown that the FCN based approach adopted in this work achieves higher detection accuracy with much fewer false alarms and miss detections.

Table 2 Comparison results between different deep learning based methods for *use by* date detection.

	Miss detection	False alarm	Accuracy
Ours	1.67%	0.28%	98.20%
CTPN [6]	2.79%	16.57%	92.20%
Seglink [7]	5.71%	12.53%	93.73%

3.3 CRNN evaluation

Next, we evaluate the performance of the CRNN for *use by* date recognition. The detailed configuration of the CRNN is shown in Table 3 according to [11]. The *use by* patches from the training image are used for fine tuning the CRNN.

Fig. 10 Selective use by date image patches for fine tuning the CRNN.

Top rows in the table 3 describe the configurations in the top layers of the CRNN and similarly, bottom rows describe the ones in bottom convolutional layers. W represents the width of the input image patch. 'k', 's' and 'p' stand for kernel size, stride and padding size respectively. The fine tune of the original CRNN is impelemnted by Pytorch in Python environment, which is consistent with the programming environment for fine tuning the first FCN network.

Table 3 The detailed configuration of the CRNN

Type	Configurations
Transcription	-
Bidirectional-LSTM	♯hidden units:256
Bidirectional-LSTM	♯hidden units:256
Map-to-Sequence	-
Convolution	♯maps:512,k:2×2,s:1,p:0
MaxPooling	Window:1 × 2, s:2
BatchNormalization	-
Convolution	♯maps:512,k:3×3,s:1,p:0
BatchNormalization	-
Convolution	♯maps:512,k:2×2,s:1,p:0
MaxPooling	Window:1 × 2, s:2
Convolution	♯maps:256,k:2×2,s:1,p:0
Convolution	♯maps:256,k:2×2,s:1,p:0
MaxPooling	Window: 2 × 2, s:2
Convolution	♯maps:128,k:3×3,s:1,p:1
MaxPooling	Window: 2 × 2, s:2
Convolution	♯maps:64,k:3×3,s:1,p:1
Input	W × 32 gray-scale image

Fig. 11 The *use by* date recognition results on different food packages.

SGD algorithm in the Pytorch libraries is applied for fine tuning the network exploiting *use by* date image patches extracted from the training dataset. Examples of use by date region recognitions for different food package images by the fine tuned CRNN are shown in Fig. 12. This figure shows that different use by date can be successfully recognized based on detected ROIs (enclosed by rectangles) by the first network for different food package images even under challenging scenarios, such as the use by date is inclined (d), the use by date region is affected by lighting (e) and printing quality is low with parts of use by date missing (f).

Moreover, we have compared the recognition performance of CRNN with other methodologies, including the most widely applied OCR tool Tesseract OCR [18] as well as another fine tuned four-stage text recognition network proposed in [12] (denoted as TPS-ResNet-BiLSTM-Att) which achieves the best performance on ICDAR2013 focused scene text and ICDAR2019 ArT

Table 4 Comparisons of *use by* date recognition by different methodologies.

Iteration	Tesseract OCR	TPS-ResNet-BiLSTM-Att network in [12]	Ours
5000	37.73%	94.01%	93.13%
9000	29.97%	94.01%	94.42%
50000	26.95%	93.59%	93.06%
100000	31.12%	94.57%	95.44%

datasets. The FCNs with different training iterations are firstly applied to extract the *use by* date regions and different recognition methodologies are exploited to recognize the *use by* date on these regions. The recognition accuracies are presented in Table 4. We can see that the deep leaning based approaches (four stage network and CRNN) greatly outperforms Tesseract OCR. The performance of the TPS-ResNet-BiLSTM-Att approach proposed in [12] achieves very similar accuracies as the fine tuned CRNN used in this work. However, the TPS-ResNet-BiLSTM-Att network in [12] is very complicated with much more parameters (48.7×10^6 parameters compared with 8.3×10^6 parameters for CRNN), thus leading to both higher storage cost for saving the network model (195.9MB required for saving the TPS-ResNet-BiLSTM-Att model and only 33.3MB is required for saving the CRNN model) and higher training/testing costs. So, the CRNN network is preferred for our task considering its high accuracy and low storage/computational costs.

4 Conclusions

In this work, we have proposed a dual deep neural network based deep learning system, for automatically recognising *use by* date information on food package images. The system composes two networks: FCN and CRNN, which are fine tuned from dealing with texts to detect/recognise *use by* date. The first FCN network identifies the region of the use by date while recognition is performed by the CRNN based on the identified ROI. The proposed system can successfully detect/recognize the *use by* date based on a variety of food package images with different colours/textures, even under low image qualities and outperforms other deep learning based methods. In the future, this system will be extended to recognize more types of information on the food package, such as ingredients introductions.

References

1. Eurostat,http://ec.europa.eu/eurostat/statistics-explained/index.php/Manufacturing statistics - NACE Rev. 2 (2014)
2. WHO FAO,http://www.fao.org/docrep/012/a1552e/a1552e00.html (2009)
3. Mori, S., Suen, C., Yamamoto, K.: Historical review of OCR research and development. Proceedings of the IEEE. 80(7), 1029-1058 (1992)
4. Epshtein, B., Ofek, E., Wexler, Y.: Detecting text in natural scenes with stroke width transform. Computer Vision and Pattern Recognition (CVPR), San Francisco, CA, USA. (2010)
5. Chen, H., Tsai, S., Schroth, G., Chen, D., Grzeszczuk, R., Girod, B.: Robust text detection in natural images with edge-enhanced Maximally Stable Extremal Regions. 18th IEEE International Conference on Image Processing (ICIP), Brussels, Belgium. (2011)
6. Tian, Z., Huang, W., He, T., He, P., Qiao., Y.: Detecting Text in Natural Image with Connectionist Text Proposal Network. 14th European Conference Computer Vision (ECCV), Amsterdam, The Netherlands. (2016)
7. Shi, B., Bai, X., Belongie, S.: Detecting Oriented Text in Natural Images by Linking Segments. International conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA. (2017)
8. Zhou, X., Yao, C., Wen, H., Liang, J.: EAST: An Efficient and Accurate Scene Text Detector. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, Hawaii, USA.(2017)
9. Bishop, C.: Pattern Recognition and Machine Learning. Springer. (2006)
10. Jaderberg, M., Vedaldi, A., Zisserman, A.: Deep features for text spotting. European Conference on Computer Vision, Zurich, Switherland. (2014)
11. Shi, B., Bai, X., Yao, C.: An End-to-End Trainable Neural Network for Image-Based Sequence Recognition and Its Application to Scene Text Recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI).39(11), 2298-2304 (2017)
12. Baek, J., Kim, G., Lee, J., Park, S., Han., D.,Yun, S., Oh, S., Lee, H.: What Is Wrong With Scene Text Recognition Model Comparisons? Dataset and Model Analysis. International Conference on Computer Vision (ICCV), Seoul, Korea. (2019)
13. Keiss, K., Khoshgoftaar, T., Wang, D.: A Survey on Transfer Learning. Journal of Big Data. 3(9), 1-40 (2016)
14. Kim, K., Hong, S., Roh, B., Cheon, Y., Park, M.: PVANET: Deep but lightweight neural networks for realtime object detection. arXiv preprint arXiv:1608.08021. (2016)
15. Simonyan, K., Zisserman, A.: Very Deep Convolutional Networks for Large-Scale Image Recognition. International Conference on Learning Representations. (2015)
16. Goodfellow, I., Bengio, Y., Courville, A.: Deep Learning. MIT. (2016)
17. Kingma, D., Ba, L.: Adam: A Method for Stochastic Optimization. International Conference on Learning Representations (ICLR), San Diego, CA. (2015)
18. Tesseract-ocr,https://github.com/tesseract-ocr/tesseract (2018)